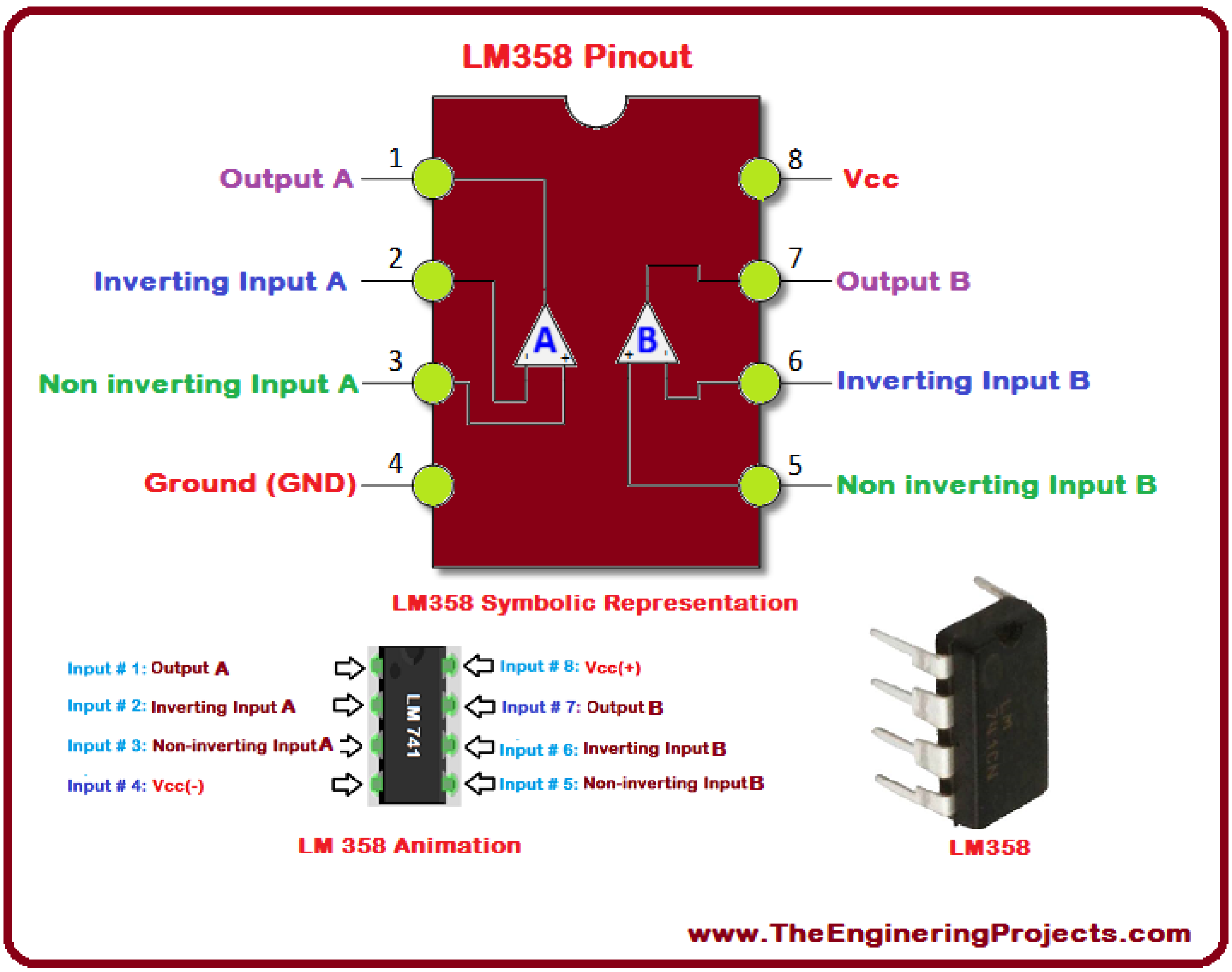
LM358 Operational Amplifier

The LM358 is a popular dual operational amplifier that includes 8 pins:

1. Output A	2. Inverting Input A
3. Non-Inverting Input A	4. GND
5. Non-Inverting Input B	6. Inverting Input B
7. Output B	8. VCC (Power Supply)



Basic Structure of LM358:

The LM358 is a dual operational amplifier with two independent amplifier circuits. Each amplifier has three pins: a non-inverting input, an inverting input, and an output. The 8 pins are clearly labeled with two sets (A and B), along with GND and VCC.

A voltage divider circuit uses two or more resistors (or other resistive elements) to divide the input voltage into several parts. By connecting resistors in series, the input voltage is proportionally distributed across the resistors, resulting in different output voltages.

Basic Principle:

The basic principle of a voltage divider is based on the series connection of resistors. For two series resistors R1 and R2, the input voltage Vin is divided into voltage drops V1 and V2 across the resistors, such that Vin = V1 + V2.

Circuit Implementation:

1: DHT11 Connection

- Connect VCC to Arduino's 5V to provide power to the sensor.

- Connect GND to Arduino GND.

- The DATA pin needs to be connected differently: connect the DATA pin to the non-inverting input pin 3 of the first op-amp in LM358.

- For the LM358, connect pin 4 to Arduino GND and pin 8 to Arduino VCC.

- Pin 1 (output A) should be connected to Arduino analog pin A0, as we are dealing with an analog signal.

- Connect pin 1 to Arduino A0.

- For pin 2 (inverting input), we will create a voltage divider using two 10kΩ resistors, resulting in a midpoint voltage of 2.5V.

Modifications for Specific Amplification:

To achieve a specific amplification factor, we need to consider the ratio of the two resistors R_f and Rin. Suppose we want a tenfold amplification; we would choose R_f as 10kΩ and Rin as 1kΩ. Everything else remains the same.

- Connect Rin from pin 2 to GND.

- Connect R_f from pin 1 to pin 2. Pin 1 is the output (A).

By choosing 10kΩ and 1kΩ resistors, we establish the gain based on their ratio.

Voltage Divider Formula:

Assuming R1 = R2 = 10kΩ and Vin = 5V, Vout = Vin * R2/(R1+R2) = 2.5v

Amplification Formula: Av = Rf / Rin = 10k / 1k = 10

Operational Amplifier Working Principle:

- Input Signal (Pin 3): The first op-amp in the LM358 amplifies the output from the DHT11 sensor.
- Output Signal (Pin 1): The amplified signal from the LM358 is connected to Arduino analog pin A0.

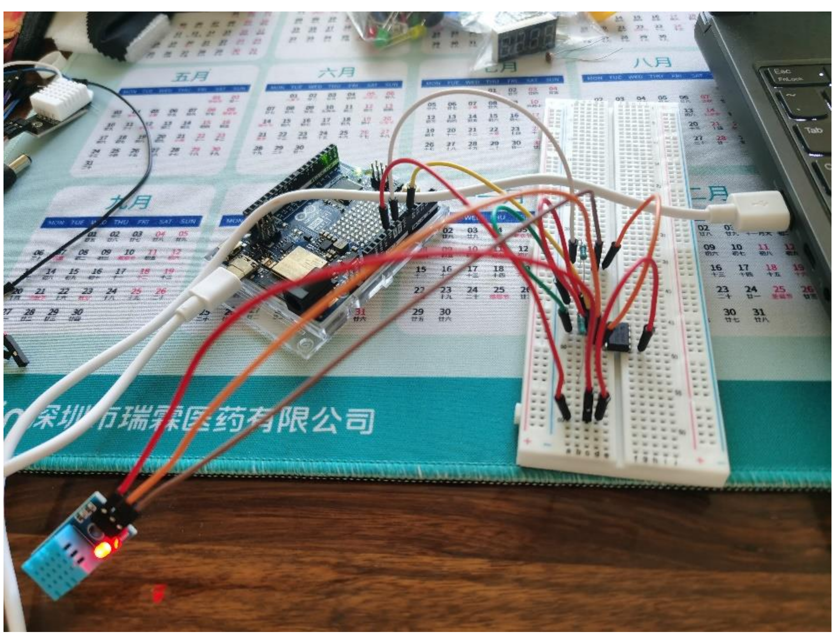
Code

```
#define AMP_PIN A0 // Define the pin connected to the DHT sensor as Analog A0

int ampValue = analogRead(AMP_PIN);

float voltage = ampValue * (5.0 / 1023.0);
```

These two lines of code read the analog value and convert it to voltage. The conversion formula 5 /1023 represents the voltage corresponding to each unit of the ADC value, where 5 is the reference voltage of the Arduino, and 1023 is the maximum value of the ADC. This shows the variation in the amplified analog signal and displays it on the Arduino.



two 10kΩ resistors

```
Output    Serial Monitor ×
Message (Enter to send message to 'Arduino UNO R4 WiFi' on 'COM8')
Amplified Signal: 22    Voltage: 0.11
Amplified Signal: 22    Voltage: 0.11
Amplified Signal: 21    Voltage: 0.10
Amplified Signal: 21    Voltage: 0.10
Amplified Signal: 21    Voltage: 0.10
Amplified Signal: 21    Voltage: 0.10
Amplified Signal: 22    Voltage: 0.11
Amplified Signal: 22    Voltage: 0.11
Amplified Signal: 21    Voltage: 0.10
Amplified Signal: 22    Voltage: 0.11
Amplified Signal: 21    Voltage: 0.10
Amplified Signal: 22    Voltage: 0.11
Amplified Signal: 21    Voltage: 0.10
Amplified Signal: 22    Voltage: 0.11
Amplified Signal: 20    Voltage: 0.10
```

10kΩ and 1kΩ resistors

```
Output    Serial Monitor ×
Message (Enter to send message to 'Arduino UNO R4 WiFi' on 'COM8')
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 15    Voltage: 0.07
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 17    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
Amplified Signal: 16    Voltage: 0.08
```

DHT11 Sensor

The DHT11 sensor outputs a digital signal. It is specifically designed to measure temperature and humidity and transmits the measurements as a digital signal to a microcontroller or other receiving devices. The DHT11 does not provide analog output, only digital data packets, simplifying design since the receiving device only processes digital signals without the need for analog-to-digital conversion.

Detect Analog Signal Changes:

Using a photoresistor can help detect changes in analog signals. Connect a photoresistor and a fixed resistor in series to form a voltage divider, then connect the output of the voltage divider to the non-inverting input of the LM358 operational amplifier.

Specific Steps:

- Connect one end of the photoresistor to the Arduino's 5V power.

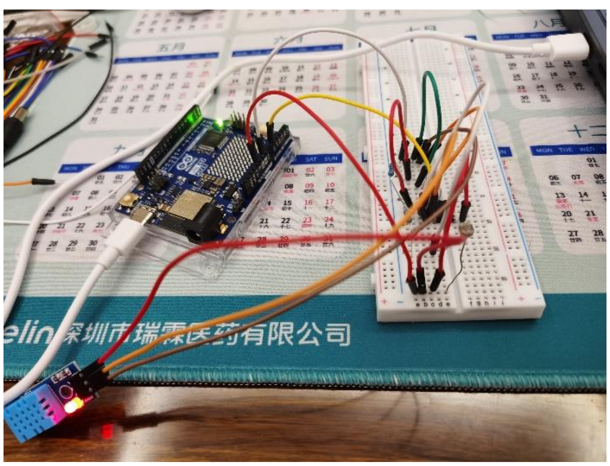
- Connect the other end of the photoresistor to a 10kΩ fixed resistor and pin 3 of the LM358.

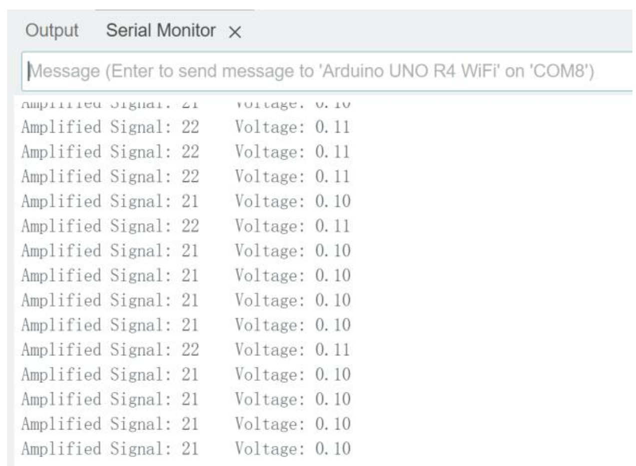
- Connect the other end of the 10kΩ resistor to Arduino GND.

- Connect the midpoint of the photoresistor and resistor to pin 3 of the LM358.

- Connect pin 2 of the LM358 to GND through a 10kΩ resistor and pin 1 of the LM358 through another 10kΩ resistor.

- Connect pin 1 to Arduino A0, pin 4 to Arduino GND, and pin 8 to Arduino 5V.






```
#include <DHT.h> // Include DHT library for communication with the DHT sensor

#define AMP_PIN A0 // Define the pin connected to the DHT sensor as Analog A0
#define DHTTYPE DHT11 // Define the type of DHT sensor as DHT11

DHT dht(AMP_PIN, DHTTYPE); // Create a DHT sensor object, specifying the pin and
sensor type
```

```
// For the photoresistor
const int sensorPin = A0; // Define the constant sensorPin, connected to input
pin A0

void setup() {
  Serial.begin(9600); // Initialize serial communication at 9600 baud
}

void loop() {
  int ampValue = analogRead(sensorPin); // Read the analog signal value from
sensorPin and store it in ampValue
  // The analogRead function returns an integer value between 0 and 1023,
representing the voltage from 0 to 5V.

  float voltage = ampValue * (5.0 / 1023.0); // Convert the analog value to
actual voltage
  // The range of analog input is 0 to 1023, corresponding to a voltage range of
0 to 5V. Convert the analog value to actual voltage.

  Serial.print("Amplified Signal: "); // Print the string "Amplified Signal: "
  Serial.print(ampValue); // Print the value of ampValue
  Serial.print("\tVoltage: "); // Print the string "Voltage: "
  Serial.println(voltage); // Print the value of voltage
  delay(1000); // Wait for 1 second before continuing the loop
}
```